

LAPORAN PRAKTIKUM DATA STRUCTURE

LINKED LIST

Dosen Pengampu:

H. Fatchurrochman,M.Kom

Asisten Praktikum:

Fillah Anjany 230605110033



Oleh :

Muhammad Alif Mujaddid

240605110082

Jurusan Teknik Informatika Fakultas Sains dan Teknologi

UIN Maulana Malik Ibrahim Malang

2025

Kesimpulan

Kesimpulan yang diperoleh dari pembahasan praktikum kali ini adalah:

1. Tentang perbandingan 4 macam struktur linked list, yaitu linier singly-linked list, linier doubly-linked list, circular singly-linked list, dan circular-doubly linked list.

Linear Singly Linked List merupakan struktur paling sederhana dari linked list. Setiap node hanya memiliki satu pointer yang menunjuk ke node berikutnya. Akses data dilakukan searah dari depan ke belakang, sehingga traversal hanya bisa dilakukan satu arah. Struktur ini cukup efisien dalam penggunaan memori, tetapi kurang fleksibel jika diperlukan akses ke arah sebelumnya.

Linear Doubly Linked List memiliki dua pointer pada setiap node, yaitu ke node berikutnya dan node sebelumnya. Hal ini memungkinkan traversal dilakukan dua arah, sehingga lebih fleksibel dibanding singly linked list. Operasi penyisipan maupun penghapusan juga lebih mudah karena dapat langsung mengakses node sebelumnya. Namun, struktur ini membutuhkan ruang memori lebih besar karena setiap node harus menyimpan dua pointer.

Circular Singly Linked List memiliki ciri khusus yaitu node terakhir menunjuk kembali ke node pertama, sehingga membentuk lingkaran. Dengan demikian, traversal dapat dilakukan secara terus-menerus tanpa perlu mengetahui panjang list. Struktur ini cocok untuk aplikasi yang membutuhkan akses berulang secara melingkar, namun tetap hanya bisa diakses satu arah.

Circular Doubly Linked List menggabungkan kelebihan dari doubly linked list dan circular linked list. Node terakhir akan menunjuk ke node pertama, sementara node pertama juga menunjuk ke node terakhir, sehingga traversal dapat dilakukan dua arah tanpa ujung. Struktur ini sangat fleksibel untuk berbagai operasi seperti penyisipan, penghapusan, dan traversal, tetapi membutuhkan memori lebih banyak karena setiap node menyimpan dua pointer.

2. Tentang implementasi Stack pada Linked list

Stack dengan Linked List diimplementasikan dengan prinsip LIFO (Last In, First Out), yaitu elemen terakhir yang masuk akan menjadi elemen pertama yang keluar. Dalam implementasinya, setiap elemen stack direpresentasikan sebagai node pada linked list. Operasi push dilakukan dengan menambahkan node baru di awal (head) linked list, sehingga elemen tersebut menjadi puncak (top) stack. Sedangkan operasi pop dilakukan dengan menghapus node pada posisi head, lalu top berpindah ke node berikutnya. Dengan cara ini, linked list dapat digunakan untuk membentuk struktur stack tanpa harus khawatir dengan keterbatasan ukuran array, karena jumlah node dapat bertambah atau berkurang secara dinamis sesuai kebutuhan.

3. Tentang implementasi Queue pada Linked list

Queue dengan Linked List diimplementasikan dengan prinsip FIFO (First In, First Out), yaitu elemen pertama yang masuk akan menjadi elemen pertama yang keluar. Dalam implementasinya, setiap elemen queue direpresentasikan sebagai node pada linked list. Operasi enqueue dilakukan dengan menambahkan node baru di bagian belakang (rear) linked list, sedangkan operasi dequeue dilakukan dengan menghapus node di bagian depan (front). Untuk memudahkan pengelolaan, biasanya digunakan dua pointer, yaitu front yang menunjuk ke node paling depan, dan rear yang menunjuk ke node paling belakang. Dengan pendekatan ini, ukuran queue tidak terbatas seperti pada array, karena node dapat ditambahkan atau dihapus secara dinamis sesuai kebutuhan.